Distributed IOT and AI platform using cloud native infrastructure

Han Yu, Binbin Yu
China Mobile (Hangzhou) Information Technology Co., Ltd.
Changes 5G brings to smart home/IOT

basic architecture
up-to-date

Nationwide IOT and AI service platform cookbook

cloud native tech we used
persistent connection
large size serving model
**basic architecture**

Smart home platform, like most IOT platforms, manages a wide range of terminal devices through specific protocols. Management functions include sending commands, on-line status monitoring, message push, software updates, etc.

OpenStack and Ceph benefits:
- Easy to clone instance
- Live-migration to optimize pressure
- Friendly cloud API
High availability is not enough

Smart home platform features are constantly enriched, we need
- faster CI/CD and scale out
- microservice with separate release
- online serving with machine learning model

1 IDC, more than one AZ, multiple active-active workload
Up-to-date

Upgrades
- most service, workflow for IoT device containerization
- Helm charts for development and production (prototype of our company's Shared Services Platform)
- province-Center arch based on edge mode
Province-center is another type of edge-computing, especially in smart home/iot

- Low latency of the response to iot device
- Service offload
- Fewer active transaction in central node
- Tiny ‘AZ’
- Behavior of IOT device or smart home user’s ETL both in province and center

And benefits

- Device acts rapidly, several seconds to less than 1 second
- Continuity of most operations by failover to node autonomy
- Less CAPEX and OPEX for center facility, such as fewer mqtt broker instances
- Some critical faults caused by baremetal only affect a small number of iot devices or smart home users
- Targeted operational strategies of different province
Based on the architecture of the province-center, we are also able to continue to promote the development of open service sinking while enriching the capabilities of smart home platforms.

The init job is like Openstack-operator and Kubernetes-operator
cloud native tech we used
Customized HPA rules for service or middleware
Healthcheck and self-healing
In order to avoid load balancer becoming a bottleneck, persistent connection is a viable solution.

- System tuning: Increase memory for NAT table. NAT table helps hypervisor or docker hold the connection.
- Service Unitization: Load balancer could be divided into POD.
- Consistent Hashing: #Google Maglev / Ucloud Vortex.
- Edge-province, province-center.
large size serving model

how to distribute?

tensorflow/serving

SavedModel

edge node

aggregation

online

Dataset Marking and training

tensorboard

R & D Center

Tensorflow

Deep learning model

VM image

Docker image

Video / release / OTT
Thank you